DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Primer TP Grupal

Especificación, Implementaciones y Demostraciones

18 de mayo de 2024 — Algoritmos y Estructuras de Datos

JAC Team

Integrante	LU	Correo electrónico
Prieto, Camila Luciana	624/15	`camilalprieto@gmail.com`
Reyes Vega, Angel Guillermo	252/23	`rvangelse@gmail.com`
Romero Huisi, Juan Cruz	549/23	`juancruzromeroh@gmail.com`
Cuellar, Aaron	810/23	`aaroncuellar2003@gmail.com`

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

1. Especificación

Aclaraciones Generales: Los índices de las listas recursos, cooperan, trayectoria, apuestas, pagos, eventos representa el identificador de los individuos. Los Predicados y funciones auxiliares utilizadas en mas de un ejercicio las ubicaremos en la sección 1.6

1.1. redistribucionDeLosFrutos

`proc` `redistribucionDeLosFrutos` (in recursos: $seq\langle\mathbb{R}\rangle$, in cooperan: $seq\langle \mathsf{Bool}\rangle$) : $seq\langle\mathbb{R}\rangle$

`requiere` $\{|recursos| = |cooperan| \land$ **extrictamentePositivos**$(recursos)\}$

`asegura` $\{|res| = |recursos| \land_L ((\forall i : \mathbb{Z})(0 \leq i < |recursos|) \longrightarrow_L res[i] = (if\ (cooperan[i]\ =\ true)\ then$ **gananciaGeneral**$(|recursos|)\ else\ recursos[i] +$ **gananciaGeneral**$(|recursos|)\ fi\ )\}$

`aux` `fondoMonetarioComun` (recursos: $seq\langle\mathbb{R}\rangle$, cooperan: $seq\langle\mathsf{Bool}\rangle$) : $\mathbb{R}$ = $\sum_{i=0}^{|recursos|-1}(if\ (cooperan[i]\ =\ true)\ then\ recursos[i]\ else\ 0\ fi\ )$;

`aux` `gananciaGeneral` (personas: $\mathbb{R}$) : $\mathbb{R}$ = $\frac{\textbf{fondoMonetarioComun}(recursos,cooperan)}{personas}$;

1.2. trayectoriadeLosFrutosIndividualesALargoPLazo

`proc` `trayectoriadeLosFrutosIndividualesALargoPLazo` (inout Trayectorias: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in cooperan: $seq\langle\mathsf{Bool}\rangle$, in Apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in Pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in Eventos: $seq\langle seq\langle\mathbb{N}\rangle\rangle$)

`requiere` $\{Trayectorias = Trayectorias_0 \land |coorperan| = |Apuestas| = |Pagos| = |Trayectorias_0| = |Eventos|\}$

`requiere` $\{(\forall i : \mathbb{Z})\ ((0 \leq i < |Trayectorias_0|) \longrightarrow_L (((|Trayectorias_0| = 1) \land Trayectorias[i][0] > 0)))\}$

`requiere` $\{$**delMismoTamaño**$(Apuestas, Pagos)\}$

`requiere` $\{$**sumatoriaDeLasComponentesDeApuestas**$(Apuestas)\}$

`requiere` $\{$**eventosPosibles**$(Eventos, Apuestas)\}$

`asegura` $\{(tamañoDeTrayectoria(Eventos, Apuestas)) \land_L (\forall i, j : \mathbb{Z})\ ((0 \leq i < |Eventos| \land_L 0 \leq j < |Eventos[i]|) \longrightarrow_L$

$if(cooperan[i] = true)\ then\ Trayectorias[i][j+1] =$ **gananciaPerCapita**$(j, Trayectorias, cooperan)$
$else\ Trayectorias[i][j+1] =$ **gananciaPerCapita**$(j, Trayectorias, cooperan +$
gananciaNeta$(i, j, Pagos, Eventos, Apuestas, Trayectorias)\ fi)\}$

`pred` `tamañoDeTrayectoria` (Eventos: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$,Apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ ((0 \leq i < |Eventos|) \longrightarrow_L (|Trayectorias[i]| = (|Eventos[i]| + 1))$
}

1.3. trayectoriaExtrañaEscalera

`proc` `trayectoriaExtrañaEscalera` (in trayectoria:$seq\langle\mathbb{R}\rangle$) : Bool {

`requiere` $\{(\forall i : \mathbb{Z})((0 < i < |trayectoria| \longrightarrow_L trayectoria[i] \geq 0) \land trayectoria[0] > 0)\}$

`asegura` $\{((|trayectoria| = 1 \land res = true) \lor (|trayectoria| > 1 \land_L res = true \longleftrightarrow$ **cantDeMax**$(trayectoria) = 1)\}$

}

`aux` `cantDeMax` (in trayectoria: $seq\langle\mathbb{R}\rangle$) : $\mathbb{Z}$ = $\sum_{i=0}^{|trayectoria|-1}(if\ esMaximo(tr, i)\ then\ 1\ else\ 0\ fi)$;

`pred` `esMaximo` (trayectoria: $seq\langle\mathbb{R}\rangle$, i:$\mathbb{Z}$) {
$(i = 0 \longrightarrow trayectoria[0] > trayectoria[1]) \land$
$(i = |trayectoria| - 1 \longrightarrow trayectoria[|trayectoria| - 1] > trayectoria[|trayectoria| - 2]) \land$
$(i \neq 0 \land i \neq |trayectoria| - 1 \longrightarrow trayectoria[i] > tr[i-1] \land trayectoria[i] > trayectoria[i+1])$
}

1.4. individuoDecideSiCooperarONo

`proc` `individuoDecideSiCooperarONo` (in individuo: $\mathbb{N}$, in recursos: $seq\langle\mathbb{R}\rangle$, inout cooperan: $seq\langle\mathsf{Bool}\rangle$, in Apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in Pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in Eventos: $seq\langle seq\langle\mathbb{N}\rangle\rangle$)

`requiere` $\{cooperan = cooperan_0\ \land\ |cooperan| = |Apuestas| = |Eventos| = |Pagos|\}$

`requiere` $\{0 \leq individuo < |recursos| \land$ **extrictamentePositivos**$(recursos)\}$

`requiere` $\{(\forall i : \mathbb{Z})\ ((0 \leq i < |recursos|) \longrightarrow_L recursos[i] > 0)\}$

`requiere` $\{$**delMismoTamaño**$(Pagos, Apuestas)\}$

`requiere` $\{$**sumatoriaDeLasComponentesDeApuestas**$(Apuestas)\}$

`requiere` $\{$**eventosPosibles**$(Eventos, Apuestas)\}$

asegura $\{((|cooperan| = |cooperan_0|) \wedge (\forall i : \mathbb{Z})\ ((0 \leq i < |cooperan|) \wedge i \neq individuo \longrightarrow_L cooperan[i] = cooperan_0[i])) \wedge$
$(\exists trCoop, trNoCoop : seq\langle seq\langle \mathbb{R}\rangle\rangle)$
$((\textbf{trayectoriaValida}(trCoop, recursos, Eventos, SetAt(cooperan, individuo, true), Apuestas, Pagos) \wedge$
$(\textbf{trayectoriaValida}(trNoCoop, recursos, Eventos, SetAt(cooperan, individuo, false), Apuestas, Pagos) \wedge_L$
$(cooperan[individuo] = true \longleftrightarrow \textbf{decisionCoopera}(trCoop, trNoCoop))\}$

pred decisionCoopera (trCoop: $seq\langle seq\langle \mathbb{R}\rangle\rangle$, trNoCoop:$seq\langle seq\langle \mathbb{R}\rangle\rangle$) {
$trCoop[individuo][|Eventos|-1] \geq trNoCoop[individuo][|Eventos|-1]$
}

1.5. individuoActualizaApuesta

proc individuoActualizaApuesta (in individuo: $\mathbb{N}$, in recursos: $seq\langle \mathbb{R}\rangle$, in cooperan: $seq\langle \mathsf{Bool}\rangle$, inout Apuestas: $seq\langle seq\langle \mathbb{R}\rangle\rangle$, in Pagos: $seq\langle seq\langle \mathbb{R}\rangle\rangle$, in Eventos: $seq\langle seq\langle \mathbb{N}\rangle\rangle$)

requiere $\{Apuestas = Apuestas_0 \wedge |cooperan| = |Apuestas_0| = |Eventos| = |Pagos|\}$
requiere $\{0 \leq individuo < |recursos| \wedge \textbf{extrictamentePositivos}(recursos)\}$
requiere $\{(\forall i : Z)(0 \leq i < |recursos| \longrightarrow_L recursos[i] > 0\}$
requiere $\{\textbf{delMismoTamaño}(Pagos, Apuestas_0)\}$
requiere $\{\textbf{sumatoriaDeLasComponentesDeApuestas}(Apuestas_0))\}$
requiere $\{\textbf{eventosPosibles}(Eventos, Apuestas_0))\}$
asegura $\{((\exists Trayectorias : seq\langle seq\langle \mathbb{R}\rangle\rangle)$
$(\textbf{trayectoriaValida}(Trayectorias, recursos, Eventos, cooperan, Apuestas, Pagos)) \wedge$
$(\forall Apuestas' : seq\langle seq\langle \mathbb{R}\rangle\rangle))((|Apuestas'| = |Apuestas_0| \wedge \textbf{esApuestaValida}(individuo, Apuestas', Pagos))) \longrightarrow_L$
$(\exists Trayectorias' : seq\langle seq\langle \mathbb{R}\rangle\rangle)(\textbf{trayectoriaValida}(Trayectorias', recursos, Eventos, cooperan, Apuestas', Pagos)) \wedge_L$

$\textbf{recursosFinales}(Trayectorias, individuo, Eventos, cooperan, Pagos, Apuestas) \geq$
$\textbf{recursosFinales}(Trayectorias', individuo, Eventos, cooperan, Pagos, Apuestas')\}$

pred esApuestaValida (in individuo: $\mathbb{N}$, Apuestas:$seq\langle seq\langle \mathbb{R}\rangle\rangle$,Pagos:$seq\langle seq\langle \mathbb{R}\rangle\rangle$) {
$\textbf{delMismoTamaño}(Apuestas, Pagos) \wedge \textbf{sumatoriaDeLasComponentesDeApuestas}(Apuestas) \wedge$
$\textbf{soloCambiaApuestaDeUnIndividuo}(individuo, Apuestas, Apuestas_0)$
}

pred soloCambiaApuestaDeUnIndividuo (individuo:N,Apuestas:$seq\langle seq\langle \mathbb{R}\rangle\rangle$,Apuestas$_{0:}$$seq\langle seq\langle \mathbb{R}\rangle\rangle$){
$(\forall i : \mathbb{Z})(0 <= i < |Apuestas_0|) \longrightarrow_L ((\forall j : \mathbb{Z})(\exists individuo : \mathbb{Z})(0 \leq j, individuo < |Apuestas_0[i]|) \longrightarrow_L (j \neq individuo \wedge Apuestas_0[i] = Apuestas[j])$
}

aux recursosfinales (Trayectorias:$seq\langle seq\langle \mathbb{R}\rangle\rangle$,individuo:$\mathbb{N}$,Eventos:$seq\langle seq\langle \mathbb{R}\rangle\rangle$,cooperan,Pagos:$seq\langle seq\langle \mathbb{R}\rangle\rangle$, Apuestas:$seq\langle seq\langle \mathbb{R}\rangle\rangle$) : $\mathbb{R}$ =
$if cooperan[individuo] = true\ then\ \textbf{gananciaPerCapita}(|Eventos|, Trayectorias, cooperan)\ else$
$(\textbf{gananciaPerCapita}(|Eventos|, Trayectorias, cooperan) +$
$\textbf{gananciaNeta}(individuo, |Eventos|, Pagos, Eventos, Apuestas, Trayectorias))fi$ **;**

1.6. Predicados y Auxiliares utilizados en varios ejercicios

,

pred extrictamentePositivos (s: $seq\langle \mathbb{R}\rangle$) {
$(\forall i : \mathbb{Z})(0 < i < |s| \longrightarrow_L Apuestas[i] > 0)$
}
pred delMismoTamaño (Apuestas:$seq\langle seq\langle \mathbb{R}\rangle\rangle$, Pagos:$seq\langle seq\langle \mathbb{R}\rangle\rangle$) {
$(\forall i, j : \mathbb{Z})\ (0 \leq i, j < |Apuestas|) \longrightarrow_L (|Apuestas[j]| = |Pagos[j]| \wedge |Apuestas[i]| = |Apuestas[j]| \wedge |Pagos[i]| = |Pagos[j]|)$
}

pred sumatoriaDeLasComponentesDeApuestas (Apuestas:$seq\langle seq\langle \mathbb{R}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ ((0 \leq i < |Apuestas|) \longrightarrow_L (\sum_{j=0}^{|Apuestas|-1} s[i][j] = 1))$
}

pred eventosPosibles (Eventos:$seq\langle seq\langle \mathbb{Z}\rangle\rangle$,Apuestas:$seq\langle seq\langle \mathbb{R}\rangle\rangle$) {
$(\forall i, j, k : \mathbb{Z})((0 \leq k < |Apuestas|) \wedge 0 \leq i < |Eventos|) \wedge_L (0 \leq j < |Eventos[i]|)) \longrightarrow_L ((Eventos[i][j] \geq 0) \wedge (Eventos[i][j] < |Apuestas[k]|)))$
}

pred trayectoriaValida (Trayectorias: $seq\langle seq\langle \mathbb{R}\rangle\rangle$, recursos: $seq\langle \mathbb{R}\rangle$, Eventos: $seq\langle seq\langle \mathbb{R}\rangle\rangle$, cooperan: $seq\langle \mathsf{Bool}\rangle$, Apuestas: $seq\langle seq\langle \mathbb{R}\rangle\rangle$, Pagos: $seq\langle seq\langle \mathbb{R}\rangle\rangle$) {

$|Trayectorias| = |Eventos| \wedge ((\forall i : \mathbb{Z})\ ((0 \leq i < |recursos|) \longrightarrow_L |Trayectorias[i]| = |Eventos| + 1)\ \wedge$
$\textbf{asignacionDeRec}(Trayectorias, recursos, Eventos, cooperan, Apuestas, Pagos))$
}

`pred asignacionDeRec` (Trayectorias: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, recursos: $seq\langle\mathbb{R}\rangle$, Eventos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, cooperan: $seq\langle\mathsf{Bool}\rangle$, Apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, Pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ ((0 \leq i < |Eventos|) \longrightarrow_L (Trayectorias[i][0] = recursos[i] \wedge (\forall j : \mathbb{Z})\ ((1 \leq j \leq |Eventos| \longrightarrow_L$
$(if\ cooperan[i] = true\ then\ Trayectorias[i][j] = \textbf{gananciaPerCapita}(j-1, Trayectorias, cooperan)\ else$
$Trayectorias[i][j] = \textbf{gananciaPerCapita}(j-1, Trayectorias, cooperan) +$
$\textbf{gananciaNeta}(i, j-1, Pagos, Eventos, Apuestas, Trayectorias)\ fi\})$
}

`aux gananciaPerCapita` (j:$\mathbb{N}$, Trayectorias: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, cooperan: $seq\langle\mathsf{Bool}\rangle$) : $\mathbb{R}$ = $\frac{\sum_{i=0}^{|cooperan|-1} (if\ cooperan[i]=true\ then\ Trayectorias[i][j]\ else\ 0\ fi)}{|cooperan|}$ **;**

`aux gananciaNeta` (i:$\mathbb{N}$, j:$\mathbb{N}$, Pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, Eventos: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$, Apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, Trayectorias: $seq\langle seq\langle\mathbb{R}\rangle\rangle$) : $\mathbb{R}$ $= Trayectorias[i][j] * Pagos[i][Eventos[i][j]] * Apuestas[i][Eventos[i][j]]$ **;**

2. Implementación y demostraciones de correctitud

```
res := recursos
i   := 0
while (i < eventos.size()) do
    if eventos[i] then
        res := (res * apuestas.c) * pago.c
    else
        res := (res * apuestas.s) * pago.s
    endif
    i := i + 1
endwhile
```

Código 1: FrutoDelTrabajoIndividual

Llamaremos a:

$$\left.\begin{array}{r}\text{res:= recursos}\\ \text{i:= 0}\end{array}\right\} \rightarrow S_1 \qquad \text{y} \qquad \left.\begin{array}{r}\textbf{while } (i < eventos.size()) \textbf{ do}\\ \textbf{if} \text{ eventos[i] then}\\ \text{res := (res} * apuestas.s) * pago.s\\ \textbf{else}\\ \text{res := (res} * apuestas.s) * pago.s\\ \textbf{endif}\\ \text{i := i + 1}\\ \textbf{endwhile}\end{array}\right\} \rightarrow S_2$$

2.1. Demostración de la Correctitud del Programa:

Para demostrar que el programa es correcto, por definición debemos probar que la tripla de Hoare $\{P\}S\{Q\}$ es valida.

$\{Requiere\}S\{Asegura\} \equiv Requiere \implies wp(S, Asegura)$
$\equiv Requiere \implies wp(S1; S2, Asegura)$
$\equiv Requiere \implies wp(S1; wp(S2, Asegura))$

Vemos que S2 es un ciclo, por lo que usaremos el teorema del invariante.
Proponemos un invariante "I", un predicado "P_c", un predicado "Q_c" y esta función variante "fv"

- $P_c \equiv (res = recursos \wedge recursos > 0 \wedge i = 0)$
- $I \equiv 0 \leq i \leq |eventos| \wedge res > 0 \wedge_L (\forall j : Z)(0 \leq j < i \longrightarrow_L res = recursos * \prod_{j=0}^{i-1}(if eventos[j] Then (Pc * Ac) Else (Ps * As)))$
- $Q_c \equiv (i = |eventos| \wedge_L (\forall j : \mathbb{Z})\ (0 \leq j < |eventos| \longrightarrow_L res = recursos * \prod_{j=0}^{i-1}(if\ eventos[j]\ Then\ (P_c * A_c) Else\ (P_s * A_s))))$
- $fv \equiv |evento| - i$

El teorema del invariante demostrará la correctitud y la terminación del ciclo. Para usarlo, debemos probar que los siguientes axiomas son válidos, en función de lo propuesto previamente:

2.2. Verificación de la correctiud del ciclo

Demostración Axioma 1 del Teorema del Invariante: $(P_c \implies I)$

- Tomando valido nuestro P_c, reemplazamos los valores $(res = recursos \wedge recursos > 0 \wedge i = 0)$ en nuestro I
- $0 \leq 0 < |eventos| \wedge recursos > 0$
  $\wedge_L (\forall j : \mathbb{Z})\ (0 \leq j < 0 \longrightarrow_L recursos = recursos * \prod_{j=0}^{0-1}(if\ eventos[j]\ Then\ (P_c * A_c) Else\ (P_s * A_s))$
- $0 < |eventos| \wedge recursos > 0 \wedge_L (\forall j : \mathbb{Z})\ (False \longrightarrow_L ....)$
  Dentro del paratodo, no existe un j que cumpla la condición. Entonces tenemos un falso que implica un verdadero trivial.

Demostración Axioma 2 del Teorema del Invariante: $\{I \wedge B\}S_2\{I\}$ tenemos que probar que $(I \wedge B) \Rightarrow wp(S_2, I)$

- **Calculamos la wp(S_2,I)**
- $wp(S_2, I) \equiv wp(IfThenElse, i := i+1, I) \equiv wp(IfThenElse, wp(i := i+1, I))$
- $wp(i := i+1, 0 \leq i \leq \wedge|eventos| \wedge res > 0 \wedge_L (\forall j : Z)(0 \leq j < i \longrightarrow_L res = recursos * \prod_{j=0}^{i-1}(ifeventos[j]Then(Pc * Ac)Else(Ps * As))))$
  Luego
- $def(i+1) \wedge_L 0 \leq i+1 \leq \wedge|eventos| \wedge res > 0 \wedge_L (\forall j : Z)(0 \leq j < i+1 \longrightarrow_L res = recursos * \prod_{j=0}^{(i+1)-1}(ifeventos[j]Then(Pc * Ac)Else(Ps * As)))$
  Donde $def(i+1)$=True y reemplazamos i:=i+1 en el invariante
- $-1 \leq i \leq \wedge|eventos| - 1 \wedge res > 0 \wedge_L (\forall j : Z)(0 \leq j < i+1 \longrightarrow_L res = recursos * \prod_{j=0}^{(i}(ifeventos[j]Then(Pc * Ac)Else(Ps * As)))$
  Realizamos un despeje de las ecuaciones
- $-1 \leq i \leq |eventos| - 1 \wedge res > 0 \wedge_L (\forall j : Z)(0 \leq j < |eventos| \longrightarrow_L res = recursos * \prod_{j=0}^{(|eventos|-1}(ifeventos[j]Then(Pc * Ac)Else(Ps * As)))$
  Reemplazando i por su máximo valor posible, observamos que los índices en donde se mueven j y la productoria están bien definidos.Ahora que sabemos esto, llamaremos a este nuevo predicado Q_1 y realizaremos la prueba de correctitud del if.
- $wp(IfThenElse, Q_1) \Rightarrow def(eventos[i]) \wedge_L (eventos[i] \wedge wp(res = res * A_c * P_c, Q_1) \vee (\neg eventos[i] \wedge wp(res = res * A_c * P_c, Q_1)$
  Donde $def(eventos[i]) = true$ pues anteriormente se probo que los índices están bien definidos

  En el paso siguiente, probaremos las guardas del if, y todos los valores de los índices están bien definidos y por lo tanto dan el valor de verdad es True
- Llamamos "X" a la wp en el caso de $\widetilde{eventos}[i] = True$, e "Y" a la wp en el caso de $eventos[i] = False$

  Además, creamos las variables moño=$|\widetilde{eventos}|$ que representan la cantidad de apariciones dependiendo el caso toma la guarda
- $X \equiv wp(res = res * A_c * P_c, Q_1) \equiv (def(res) \wedge def(A_c) \wedge def(P_c)) \wedge_L (Q_1)^{res}_{res*A_c*P_c}$
- $(Q_1)^{res}_{res*A_c*P_c} \equiv -1 \leq i \leq |eventos| - 1 \wedge 0 < res * A_c * P_c \wedge_L ((\forall j : \mathbb{Z})\ (0 \leq j < i+1 \longrightarrow_L res * A_c * P_c = recursos * \prod_{j=0}^{\widetilde{(i+1)}-1}(A_c * P_c)))$
- $\equiv -1 \leq i \leq |eventos| - 1 \wedge 0 < res * A_c * P_c \wedge_L ((\forall j : \mathbb{Z})\ (0 \leq j < |eventos| - 1 \longrightarrow_L res * A_c * P_c = recursos * \prod_{j=0}^{(|\widetilde{eventos}|-1}(A_c * P_c))$
- $res * A_c * P_c = recursos * \prod_{j=0}^{(|\widetilde{eventos}|-1}(A_c * P_c)) \equiv res * A_c * P_c = recursos * (A_c * P_c)^{(|\widetilde{eventos}|-1+1)}$
  $\equiv res = recursos * (A_c * P_c)^{(|\widetilde{eventos}|-1)}$
- Probamos que sucede suponiendo el caso de $eventos[i] = true$. Cuando entramos a la productoria, si la guarda es True el "If" será siempre por "then", por lo que $res = recurso * \prod_{j=0}^{(|\widetilde{eventos}|-1}(A_c * P_c)$, que dijimos en el item anterior que equivale a $res = recursos * (A_c * P_c)^{(|\widetilde{eventos}|-1)}$ lo cual es nuestra condición "X"
- $Y \equiv wp(res = res * A_s * P_s, Q_1) \equiv (def(res) \wedge def(A_s) \wedge def(P_s)) \wedge_L (Q_1)^{res}_{res*A_s*P_s}$
- $(Q_1)^{res}_{res*A_s*P_s} \equiv -1 \leq i \leq |eventos| - 1 \wedge 0 < res * A_s * P_s \wedge_L ((\forall j : \mathbb{Z})\ (0 \leq j < i+1 \longrightarrow_L res * A_s * P_s = recurso * \prod_{j=0}^{\widetilde{(i+1)}-1} A_s * P_s)))$
- $\equiv -1 \leq i \leq |eventos| - 1 \wedge 0 < res * A_s * P_s \wedge_L ((\forall j : \mathbb{Z})\ (0 \leq j < |eventos| - 1 \longrightarrow_L res * A_s * P_s = recurso * \prod_{j=0}^{|\widetilde{eventos}|-1}(A_s * P_s))$
- $res * A_s * P_s = recurso * \prod_{j=0}^{(|\widetilde{eventos}|-1}(A_s * P_s)) \equiv res * A_s * P_s = recursos * (A_s * P_s)^{(|\widetilde{eventos}|-1+1)}$
  $\equiv res = recursos * (A_s * P_s)^{(|\widetilde{eventos}|-1)}$

- Ahora, probamos que sucede suponiendo el caso de $eventos[i] = false$. Cuando entramos a la productoria, con la guarda en False "If" sera siempre por "else", por lo que $res = recurso * \prod_{j=0}^{(|\widetilde{eventos}|-1}(A_s * P_s)$, que dijimos en el item anterior que equivale a $res = recursos * (A_s * P_s)^{(|\widetilde{eventos}|-1)}$ lo cual es nuestra condición "Y"

- **Ahora Calculamos $I \wedge B$:**

- $0 \leq i \leq |eventos| \wedge res > 0 \wedge_L (\forall j : Z)(0 \leq j < i \longrightarrow_L res = recursos * \prod_{j=0}^{i-1}(if eventos[j] Then(Pc * Ac) Else(Ps * As))) \wedge i < |eventos|$

Entonces

- $0 \leq i \leq |eventos| - 1 \wedge res > 0 \wedge_L (\forall j : Z)(0 \leq j < |eventos| - 1 \longrightarrow_L res = recursos * \prod_{j=0}^{|eventos|-2}(if eventos[j] Then(Pc * Ac) Else(Ps * As)))$

Partimos la productoria en los dos casos:

  - $0 \leq \widetilde{i \leq} \wedge |eventos| - 1 \wedge res > 0 \wedge_L (\forall j : Z)(0 \leq \widetilde{j} < |eventos| - 1 \longrightarrow_L res = recursos * \prod_{j=0}^{|\widetilde{eventos}|-2}(Pc * Ac))) \equiv recursos * (Pc * Ac)^{|\widetilde{eventos}|-1}$
  - $0 \leq \widetilde{i \leq} \wedge |eventos| - 1 \wedge res > 0 \wedge_L (\forall j : Z)(0 \leq \widetilde{j} < |eventos| - 1 \longrightarrow_L res = recursos * \prod_{j=0}^{|\widetilde{eventos}|-2}(Ps * As) \equiv recursos * (Ps * As)^{|\widetilde{eventos}|-1)}$

- Como nos coinciden las equivalencias de las dos cálculos anteriores entonces nos queda demostrado que I $\wedge B \implies wp(S_2, I)$

- **Demostración Axioma 3 del Teorema del Invariante:** $(I \wedge \neg B) \implies Q_c$

  - $0 \leq i \leq |eventos| \wedge res > 0 \wedge_L (\forall j : \mathbb{Z})\ (0 \leq j < i \longrightarrow_L res = recursos * \prod_{j=0}^{i-1}(if\ eventos[j]\ Then\ (P_c * A_c) Else\ (P_s * A_s))) \wedge \neg(i < |eventos|)$
    $Como \neg B$ es $(|eventos| \leq i)$ y de I tenemos $0 \leq i \leq |eventos|$, el único i que vale es cuando $i = |eventos|$
  - $0 < res\ \wedge i = |eventos| \wedge_L (\forall j : \mathbb{Z})\ (0 \leq j < i \longrightarrow_L res = recursos * \prod_{j=0}^{i}(if\ eventos[j]\ Then\ (P_c * A_c) Else\ (P_s * A_s)))$ Luego
  - $0 < res\ \wedge i = |eventos| \wedge_L (\forall j : \mathbb{Z})\ (0 \leq j < |evento| \longrightarrow_L res = recursos * \prod_{j=0}^{i}(if\ eventos[j]\ Then\ (P_c * A_c) Else\ (P_s * A_s))) \implies Q_c$ ✓

2.2.1. Verificacion de la terminacion del ciclo:

- **Demostracíon Axioma 1 de la terminacion del ciclo:** {I $\wedge B \wedge fv = v_0$} S_2{fv <v_0}

  - $(I \wedge B \wedge fv = v_0) \implies wp(S_2, fv < v_0)$
  - **Calculamos** $wp(S_2, fv < v_0)$
  - $wp(IfThenElse; i; |eventos| - i < v_0) \equiv wp(IfThenElse, (wp(i := i + 1, |eventos| - i < v_0))$
  - $wp(i := i + 1, |eventos| - i < v_0) \equiv def(i + 1) \wedge_L |eventos| - (i + 1) < v_0 \equiv |eventos| < v_0 + i + 1$
  - $(def(i) \wedge def(1)) \wedge_L |eventos| < v_0 + i + 1$
  - $wp(if\ eventos[i]\ then\ (res = res * A_c * P_c)\ else\ s = res * A_s * P_s, |eventos| < v_0 + i + 1)$
  - $def(eventos[i]) \wedge_L ((eventos[i] \wedge (res = res * A_c * P_c\ , |eventos| < V_0 + i + 1)) \vee (\neg eventos[i] res = res * A_s * P_s\ , |eventos| < v_0 + i + 1))$
  - Tenemos que nuestra $wp(S_2, fv < v_0) \equiv (0 \leq i < |eventos|) \wedge_L ((eventos[i] \wedge |eventos| < v_0 + i + 1) \vee (\neg eventos[i] \wedge |eventos| < v_0 + i + 1 \leq v_0 + |eventos| + 1))$
  - $0 \leq i \leq |eventos| \wedge res > 0 \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \longrightarrow_L res = recursos * \prod_{j=0}^{i-1}(if\ eventos[j]\ Then\ (P_c * A_c) Else\ (P_s * A_s))) \wedge (i < |eventos|) \wedge (|eventos| - i = v_0))$
  - **Calculamos** {I $\wedge B \wedge fv = v_0$}
  - Para cumplir con la condición del índice de I y de B, debemos hacer una pequeña modificación: $0 \leq i < |eventos|$
  - $|eventos| - i = v_0\ \wedge\ 0 \leq i < |eventos|\ \wedge\ res > 0 \wedge_L (\forall j : \mathbb{Z})\ (0 \leq j < i \wedge_L\ res = recursos * \prod_{j=0}^{i-1}(if\ eventos[j]\ Then\ (P_c * A_c)\ Else\ (P_s * A_s))$

- Ahora reemplazamos fv en $0 \leq i < |eventos|$ y obtenemos $0 \leq |eventos| - v_0 < |eventos|$, que reescribiendo un poco nos deja:
- $(0 \leq |eventos| < |eventos| + v_0)$, y en particular $(0 \leq |eventos| < |eventos| + v_0 + 1)$
- Como ambos calculos que realizamos son equivalentes entonces probamos $(I \land B \land fv = v_0) \Rightarrow wp(S_2, fv < v_0)$ ✓

- **Demostracíon Axioma 2 de la terminacion del ciclo:** $\{\text{I} \land fv \leq 0\} \implies \neg B$
  - Sabemos que $B \equiv i < |eventos|$ entonces $\neg B \equiv i \geq |eventos|$
  - $0 \leq i \leq |eventos| \land res > 0 \land_L (\forall j : Z)(0 \leq j < i \longrightarrow_L res = recursos * \prod_{j=0}^{i-1}(if eventos[j] Then (Pc * Ac) Else (Ps * As))) \land |eventos| - i \leq 0$
  - Si acomodamos fv tenemos que $|eventos| \leq i$ (lo cual es igual a $\neg B$, pero veamos como continúa)
  - $0 \leq i \leq |eventos| \land i \geq |eventos| \iff i = |eventos|$
  - $i = |eventos| \in \neg B$
  - Por lo que podemos concluir que $\{\text{I} \land fv \leq 0\} \implies \neg B$ ✓

Concluida la demostración de los axiomas, podemos tomarlos como válidos y asegurar que se cumple la tripla de hoare $\{P_c\}S_2\{Q_c\}$

Ahora retomando, $Requiere \implies wp(S1, wp(S2, Asegura)) \equiv Requiere \implies wp(S1, P_c)$

Con esto corraboramos que nuestro Pc es valido con respecto al Requiere del programa. Con esto verficamos si nuestro Requiere permite que lleguemos al ciclo con variables y condiciones validas que no rompan el ciclo

- **Demostración** $Requiere \implies P_c \equiv Requiere \implies wp(S_1, P_c)$
  - $Requiere \implies wp(res := recursos; i := 0, P_c) \equiv wp(res := recursos, wp(i := 0, i = 0 \land res = recursos \land recursos > 0))$
  - $wp(res := recursos, wp(i := 0, i = 0 \land res = recursos \land recursos > 0))$
    $\equiv def(0) \land_L Q_0^i$
    $\equiv True \land_L (0 = 0 \land res = recursos \land recursos > 0)$
    $\equiv (True \land_L True \land res = recursos \land recursos > 0) \equiv res = recursos \land recursos > 0$
  - $wp(res := recursos, res = recursos \land recursos > 0)$
    $\equiv def(recursos) \land_L Q_{recursos}^{res}$
    $\equiv def(recursos) \land_L recursos = recursos \land recursos > 0$
    $\equiv True \land_L recursos = recursos \land recursos > 0$
    $\equiv (True \land_L True \land recursos > 0) \equiv recursos > 0$
  - Ahora nos queda demostrar que $Requiere \implies recursos > 0$
  - $(apuesta_c + apuesta_s = 1 \land pago_c > 0 \land pago_s > 0 \land apuesta_c > 0 \land apuesta_s > 0 \land recursos > 0) \implies recursos > 0$
  - Tomando valido el Requiere, tenemos que $recursos > 0 \implies wp(S_1, P_c) \equiv recursos > 0$ lo cual resulta en una tautología ✓

Ahora, nos queda por demostrar que nuestro Qc es valido de acuerdo al asegura del programa

- **Demostración** $Q_c \implies Asegura$
  - $Q_c \equiv (i = |eventos| \land_L (\forall j : \mathbb{Z})\ (0 \leq j < |eventos| \longrightarrow_L res = recursos * \prod_{j=0}^{i-1}(if\ eventos[j]\ Then\ (P_c * A_c) Else\ (P_s * A_s))))$
  - $Asegura \equiv recurso * (apuesta_c * pago_c)^{\#apariciones(eventos,T)}(apuesta_s * pago_s)^{\#apariciones(eventos,F)}$
  - Si abrimos el if del Q_c nos queda que se tiene que cumplir las siguientes propiedades para el resultado dependiendo en que valor de la guarda caiga:
    - res = recursos * $\prod_{j=0}^{|evento|-1}(if\ eventos[j] = true\ Then\ (P_c * A_c) Else\ 1)$
      $\equiv recurso * (apuesta_c * pago_c)^{\#apariciones(eventos,T)}$
    - res = recursos * $\prod_{j=0}^{|evento|-1}(if\ eventos[j] = false\ Then\ (P_s * A_s) Else\ 1)$
      $\equiv recursos(apuesta_s * pago_s)^{\#apariciones(eventos,F)}$
  - Entonces se demuestra que $Q_c \implies asegura$ ✓

Finalmente, por monotonía, demostramos $Requiere \implies wp(S1; S2, Asegura)$, lo que equivale a demostrar que la tripla de Hoare $\{Requiere\}S\{Asegura\}$ es válida, por lo tanto queda demostrado que la especificacion es correcta con respecto a la implementacion, afirmando que nuestro programa es correcto.